Additional Table 2: Description of Machine learning algorithms employed

S.no	Machine learning employed	Description
1	Support Vector Machines (SVM) with kernel methods	Support Vector Machines (SVMs) are a powerful class of supervised learning algorithms primarily used for classification tasks, including binary classification. SVMs aim to find the optimal hyperplane that maximizes the margin between the two classes in the feature space. By using kernel functions, SVMs can efficiently handle non-linear decision boundaries, which can be beneficial for your dataset containing a mix of numerical and categorical features. SVMs are particularly effective when the number of features is larger than the number of samples, and they are robust to outliers and noise in the data. Additionally, SVMs can provide probabilistic outputs, which can be useful for certain applications. The choice of kernel function and hyperparameter tuning are crucial for the performance of SVM models on your binary classification problem[1].
2	Gaussian Naive Bayes (GNB)	Gaussian Naive Bayes is a simple, yet effective, probabilistic classifier that assumes independence between features. While this assumption may not always hold in real-world datasets, GNB can still perform surprisingly well, especially for binary classification tasks. GNB is known to be robust to irrelevant features and can handle both numerical and categorical variables. It serves as a useful baseline model to compare the performance of more complex algorithms in your study[2].
3	Logistic Regression	Logistic regression is a widely-used algorithm for binary classification problems. It models the probability of the binary target variable as a function of the predictor variables using a logistic sigmoid function. Logistic regression is capable of capturing linear relationships between the features and the target, making it a suitable choice for your dataset, which contains a mix of numerical and categorical variables. The interpretability of logistic regression models can also be beneficial for understanding the important factors in your classification task[3].
4	Decision Tree	Decision trees are a popular machine learning algorithm well-suited for binary classification tasks, especially when dealing with a mix of numerical and categorical features. They work by recursively partitioning the feature space into smaller regions, creating a tree-like structure of decisions. At each internal node of the tree, a decision is made based on the value of a particular feature, and this process continues until a leaf node is reached,

---

Reference:

[1] Cristianini N, Scholkopf B. Support vector machines and kernel methods: the new generation of learning machines. Ai Magazine. 2002 Sep 15;23(3):31-.

[2] Bi ZJ, Han YQ, Huang CQ, Wang M. Gaussian naive Bayesian data classification model based on clustering algorithm. In2019 International Conference on Modeling, Analysis, Simulation Technologies and Applications (MASTA 2019) 2019 Jul (pp. 396-400). Atlantis Press.

[3] Landwehr N, Hall M, Frank E. Logistic model trees. Machine learning. 2005 May;59:161-205.

		which represents the final binary prediction. Decision trees are known for their interpretability, as the decision-making process can be easily visualized and understood. They are capable of capturing complex, non-linear relationships in the data, making them a robust choice for your binary classification problem[4].
5	Random Forest	Random Forest is an ensemble learning method that combines multiple decision trees to improve the overall predictive performance and robustness of the model. It works by training several decision trees on random subsets of the training data and random subsets of the features. During prediction, the output of the individual trees is aggregated (e.g., by majority voting) to make the final binary classification. The diversity of the trees in the ensemble helps to reduce overfitting and improve the model's generalization ability, even in the presence of both numerical and categorical features. Random Forest is known for its ability to handle high-dimensional data, missing values, and a mix of feature types, making it a suitable choice for your binary classification task[5].
6	eXtreme Gradient Boosting (XGBoost)	XGBoost is a highly efficient and scalable implementation of gradient boosting, a powerful ensemble technique. XGBoost is particularly well-suited for tabular data with a mix of numerical and categorical features, as is the case in your study. It automatically handles categorical variables by encoding them into a numerical representation, allowing it to capture complex interactions between the features. XGBoost's ability to learn non-linear patterns makes it a strong candidate for binary classification tasks with potentially complex decision boundaries[6].
7	AdaBoost	AdaBoost is an ensemble learning algorithm well-suited for binary classification problems. It works by iteratively training weak learners, such as decision stumps (single-feature decision trees), and then combining them into a strong classifier. AdaBoost focuses on the misclassified instances in each iteration, adjusting the weights of the samples to improve the performance on harder-to-classify examples. This makes AdaBoost robust to noise and outliers, which is beneficial for datasets with binary target variables. Additionally, AdaBoost can handle both numerical and categorical features effectively, making it a suitable choice for your study[7].
8	K-nearest neighbors (KNN) algorithm	KNN is a non-parametric, instance-based learning algorithm that can effectively handle both numerical and categorical features. It makes predictions based on the majority vote of the k nearest neighbors in the feature space, allowing it to capture complex decision boundaries. KNN's

[4] Navada A, Ansari AN, Patil S, Sonkamble BA. Overview of use of decision tree algorithms in machine learning. In2011 IEEE control and system graduate research colloquium 2011 Jun 27 (pp. 37-42). IEEE.

[5] Rigatti SJ. Random forest. Journal of Insurance Medicine. 2017 Jan 1;47(1):31-9.

[6] Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H, Chen K, Mitchell R, Cano I, Zhou T. Xgboost: extreme gradient boosting. R package version 0.4-2. 2015 Aug 1;1(4):1-4.

[7] Schapire RE. Explaining adaboost. InEmpirical Inference: Festschrift in Honor of Vladimir N. Vapnik 2013 Oct 9 (pp. 37-52). Berlin, Heidelberg: Springer Berlin Heidelberg.

		resilience to noise and ability to work with mixed data types make it a relevant choice for your binary classification task[8].
9	CatBoost Classifier	CatBoost is a gradient boosting framework designed to handle categorical features seamlessly, without the need for manual pre-processing. This makes it a compelling choice for your study, as your dataset contains a mix of numerical and categorical variables. CatBoost automatically encodes the categorical features in an optimal way, allowing it to capture the underlying relationships between the predictors and the binary target variable. Its robust handling of mixed data types and strong performance on tabular datasets make CatBoost a well-suited algorithm for your classification problem[9].
10	Feedforward Neural Network (FFNN)	Neural networks, such as the feedforward neural network, are powerful, flexible models that can learn intricate, non-linear relationships in the data. The ability of FFNN to handle both numerical and categorical features, as well as its capacity to approximate a wide range of functions, make it a suitable candidate for your binary classification problem. Neural networks can potentially capture complex patterns in your dataset that may not be easily identified by other algorithms[10].

[8] Kramer O, Kramer O. K-nearest neighbors. Dimensionality reduction with unsupervised nearest neighbors. 2013:13-23.

[9] Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A. CatBoost: unbiased boosting with categorical features. Advances in neural information processing systems. 2018;31.

[10] Sarang P. Artificial neural networks with TensorFlow 2. Apress: Berkeley, CA, USA. 2021.